PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/18, 12/54

A1

(11) International Publication Number: WO 97/26737

(43) International Publication Date: 24 July 1997 (24.07.97)

(21) International Application Number: PCT/US97/00488

(22) International Filing Date: 10 January 1997 (10.01.97)

(30) Priority Data:
60/009,919 16 January 1996 (16.01.96) US

(71) Applicant: ASCOM NEXION INC. [US/US]; 289 Great Road, Acton, MA 01720 (US).

(72) Inventors: HUNT, Douglas, H.; 43 Pine Street, Sudbury, MA 01776 (US). NAIR, Raj, Krishnan; 284 Great Road, Acton, MA 01720 (US).

(74) Agents: LEBOVICI, Victor, B. et al.; Weingarten, Schurgin, Gagnebin & Hayes L.L.P., Ten Post Office Square, Boston, MA 02109 (US).

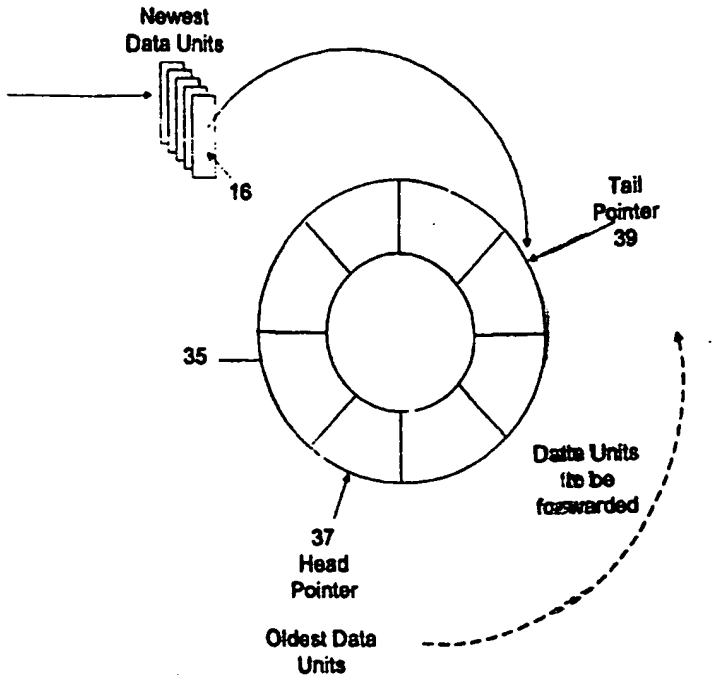
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: A RELIABLE AND FLEXIBLE MULTICAST MECHANISM FOR ATM NETWORKS

(57) Abstract

A method is disclosed for facilitating multicast operation in a network in which a data unit is multicast from a root node to a plurality of leaves via a plurality of branching point nodes in response to feedback processed at each branching point node. At least one cell forwarding technique is selected from a plurality of cell forwarding techniques at the respective branching point nodes. The cell forwarding techniques facilitate multicast operation by controlling forwarding and discard of multicast cells. The forwarding techniques are realized via use of a ring buffer (35) in which cells are stored prior to forwarding. Manipulating head (37) and tail (39) pointers associated with the ring buffer (35) allows for a plurality of desirable forwarding techniques.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgystan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic			SE	Sweden
CG	Congo	KR	Republic of Korea	SG	Singapore
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LR	Liberia	SZ	Swaziland
CS	Czechoslovakia	LT	Lithuania	TD	Chad
CZ	Czech Republic	LU	Luxembourg	TG	Togo
DE	Germany	LV	Latvia	TJ	Tajikistan
DK	Denmark	MC	Monaco	TT	Trinidad and Tobago
EE	Estonia	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	UG	Uganda
FI	Finland	ML	Mali	US	United States of America
FR	France	MN	Mongolia	UZ	Uzbekistan
GA	Gabon	MR	Mauritania	VN	Viet Nam

A RELIABLE AND FLEXIBLE MULTICAST MECHANISM FOR ATM NETWORKS

CROSS-REFERENCE TO RELATED APPLICATIONS

A claim of priority is made to U.S. Provisional Patent Application No. 60/009,919 entitled A RELIABLE AND FLEXIBLE MULTICAST MECHANISM FOR ABR SERVICE IN ATM NETWORKS, filed January 16, 1996.

STATEMENT REGARDING GOVERNMENT OWNERSHIP

Not applicable.

FIELD OF THE INVENTION

The present invention is generally related to Asynchronous Transfer Mode networks, and more particularly to multicasting within such networks.

BACKGROUND OF THE INVENTION

The advent of high-speed, cell-based, connection-oriented Asynchronous Transfer Mode ("ATM") networks creates a need for a reliable and flexible multicast mechanism that can support traditional LAN-based applications. Multicast functionality is required for implementation of "webcasting," routing, address-resolution and other inter-networking protocols. One of the early contributions to the ATM forum, "LAN Emulation's Needs For Traffic Management" by Keith McCloghrie, ATM Forum 94-0533, described multipoint connections in support of multicasting as one of the high-level requirements for LAN emulation. Such

requirements may be viewed as including at least the same level of performance from an emulated LAN as from a traditional LAN in all respects, including multicast capability.

Known techniques for implementing multicast generally fall into two categories: "slowest-leaf wins" and "best-effort delivery." Slowest-leaf wins implies that the slowest leaf of the multicast connection determines the progress of the entire connection. While this technique prevents cell loss, it may be undesirable from the point of view of public-carrier networks where it is important to avoid allowing an arbitrary end-system from controlling the performance of the network. "Best-effort delivery" implies that cells are dropped for leaves that are unable to maintain a predetermined pace. While this technique prevents an arbitrary leaf from controlling the performance of the network, dropping cells in order to maintain performance may also be undesirable, as for example with loss sensitive transfers such as computer data transmission. While these techniques might be suitable for some multicast applications, neither technique provides a satisfactory multicast mechanism for the broad range of applications encountered in high-speed networks.

SUMMARY OF THE INVENTION

In a network where a data unit is multicast in a connection from a root node to a plurality of leaves via a plurality of branching point nodes, at least one forwarding technique selected from a plurality of forwarding techniques is implemented at each branching point node. The forwarding techniques facilitate multicast operation by controlling forwarding of multicast data units, and different connections may employ different forwarding techniques. Possible forwarding techniques that may be employed include a Prevent-Loss (PL) technique, a Prevent-Loss for Distinguished Subsets (PL(n)) technique, a Prevent-Loss for Variable Subset (p/n) technique, a K-Lag technique and a K-Lead technique.

Each branching point node includes a forwarding buffer which may be modeled as a ring buffer with two pointers: a head pointer and a tail pointer. A buffer system is also provided and data units are stored in the buffer system upon receipt before entering the ring buffer. The tail pointer points to the first received and buffered data unit in the series of the most recently received data units from upstream that has not yet been forwarded to any branch. The head pointer points to the oldest data unit that needs to be forwarded downstream to any branch. Thus, the data units stored in the ring buffer between the head and tail pointers need to be forwarded to one or more branches. The tail pointer advances by one buffer in

a counter-clockwise direction with the forwarding of the most recent data unit in the ring and the arrival of the next most recent data unit that has not yet been forwarded to any branch. The head pointer advances as needed to indicate the current oldest data unit in the ring buffer. The advancement of the head pointer depends on the particular forwarding technique chosen from a range of possible techniques, each of which provides a different service guarantee. Each buffer, i, in the ring has an associated counter called a reference count, r(i), that counts the number of branches to which the data unit in the buffer must be forwarded. If the index (i) increases from head pointer to tail pointer, then one may observe that r(i) is a monotonically increasing function. Each time the tail pointer moves to a new buffer, the corresponding reference count is set to n, the number of downstream branches being serviced by that reference counter. Each time the data unit is forwarded to one of those branches, the reference counter is decremented by one. When the reference counter corresponding to the buffer at the position of the head pointer reaches 0, the head pointer is advanced in the counter-clockwise direction to the next buffer with a non-zero reference counter. Under no circumstance is the head pointer advanced beyond the tail pointer. Alternatively, there can be multiple instances of head and tail pointers with their associated reference counters where each instance can service a different disjoint subset of

branches each with a possibly different service guarantee. In this case, it is important to advance the tail pointer only in relation to the head pointers from other subsets to avoid violating the service guarantees associated with those subsets of branches. In another alternative embodiment there can be a per-branch counter rather than a per-data unit buffer counter. The general principles of the mechanism remain the same. Different service guarantees can be supported by the above mechanism such as a Prevent-Loss (PL) technique, a Prevent-Loss for Distinguished Subsets (PL(n)) technique, a Prevent-Loss for Variable Subset (p/n) technique, a K-Lag technique, a K-Lead technique and other guarantees.

The Prevent-Loss (PL) technique prevents data loss within a connection. In particular, the Prevent-Loss technique ensures that each data unit that is received is forwarded to each branch. However, if a branch performs poorly, the effect of this poor performance may eventually propagate towards the root node and thereby affect all of the branches.

The Prevent-Loss (PL) technique is realized by ensuring that the tail pointer never overtakes the head pointer in all circumstances. The head pointer can advance only after its data unit has been forwarded to each branch, i.e., only when the reference count has decreased to zero. This implements the Prevent-Loss technique.

The Prevent-Loss for Distinguished Subsets technique

guarantees delivery of the multicast data unit to predetermined subsets of branches in the multicast connection. More particularly, transmission to a distinguished subset of branches is made according to the Prevent-Loss technique described above. Hence, there is no data loss in this distinguished subset. A non-distinguished subset of branches consisting of the remaining branches in the connection, is not guaranteed delivery of the multicast data unit. Hence, the distinguished subsets of branches are insulated from possible poor performance by branches in the non-distinguished subset of branches.

The Prevent-Loss for Distinguished Subsets technique is implemented by using a separate set of reference counters for each distinguished subset of branches. The head pointer is advanced only after all members of the distinguished subset have been served. This makes it possible, for example, to provide a lossless service to the members of the distinguished subset of branches.

The K-Lag technique is realized by ensuring that the head pointer is advanced together with the tail pointer such that it is no more than a distance of K from the tail pointer. If the tail pointer is K data unit buffer positions ahead of the head pointer in the counter-clockwise direction, the data unit at the tail pointer is forwarded to any branch and the tail pointer advanced along with the head pointer one buffer

position counter-clockwise, i.e., the data unit in the buffer position of the original head pointer is no longer guaranteed to be forwarded. It is possible that the data unit at the previous head pointer positions may yet be forwarded before being overwritten by that at the tail pointer. In an alternative embodiment, such a data unit can be deleted and not forwarded any more. It should also be noted that K must be at least 1. Different values of K give rise to different levels of service.

In a variation of the K-Lag technique, K-EPD (early packet discard), each branch is allowed to lag by up to K data units, as set by an upper memory bound associated with the branch. If the tail pointer is K buffer positions ahead of the head pointer in the counter-clockwise direction and the head pointer is pointing to a data unit in the middle of the frame, then the head pointer is advanced up to the End Of Frame data unit or one position before the tail pointer, to prevent forwarding of all data units associated with the respective frame and thus avoid the waste of network bandwidth and resources.

In the K-Lead technique, the fastest branches cannot be ahead of any other branches by more than K data units. The K-Lead technique is realized by ensuring that the tail pointer is not advanced more that a distance of K from the head pointer. If the tail pointer is K buffer positions ahead of the head pointer in the counter-clockwise direction, the data unit at

the tail pointer may not be forwarded to any branch.

In the Prevent-Loss for Variable Subset (p/n) forwarding technique there is no data loss for p out of n branches. This is achieved as a variant to the PL technique. In this technique, the tail pointer is allowed to advance beyond the head pointer, as long as the corresponding reference counter is less than n-p. This ensures that a given data unit copy is delivered to at least p branches. It is possible that the unincluded branch may overflow at some point. However, the performance of the possibly overflowing branch does not affect the performance of branches in the subset of p branches. Hence, the technique provides protection for a subset of branches where the members of the subset are determined contemporaneously with the forwarding calculation.

BRIEF DESCRIPTION OF THE DRAWING

The invention will be more fully understood in view of the following Detailed Description of the Invention and Drawing, of which:

Fig. 1 is a block diagram of an ATM switch for providing reliable and flexible multicast;

Fig. 2 is a block diagram of a network topology which illustrates forwarding techniques;

Fig. 3 illustrates a ring buffer;

Fig. 4 is a flow diagram which illustrates a Prevent-Loss for Variable Subset (p/n) technique; and

Fig. 5 is a diagram which illustrates the K-EPD technique.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 illustrates a switch 10 for achieving reliable and flexible multicast functionality for an Asynchronous Transfer Mode ("ATM") network. The switch 10 has a plurality of input ports 12 and output ports 14, each of which may include an associated buffer such as a First-In First-Out ("FIFO") memory 16, 17, respectively. The input ports 12 and output ports 14 are interconnected by a switch fabric 18 such that a data unit 19, as for example a cell, frame or packet, entering any of the input ports 12 may be transmitted through the switch fabric 18 to any of the output ports 14. In particular, when one input port is using one or more output ports, any other input can use any unused output ports. The output FIFOs 17 may be sized to cope with latency such that the switch is non-blocking, i.e., each FIFO 16 at the input side may be sized to achieve a target bandwidth utilization given the round trip latencies affecting the control loop, and each output FIFO 17 may be sized to cope with such latencies in the data forwarding path. Further, the switch is provided with hardware multicast capability.

Referring now to Fig. 2, the multicast forwarding techniques will be described with regard to the illustrated

tree network topology. The tree topology includes a root node 20, a plurality of branching point nodes ("branches") 22, 24 and a plurality of end-nodes, end-systems or leaves 26, 28, 30, 32. The end-systems may be hosts, routers, or switches that terminate the tree. The branching nodes 22, 24, which stem from the root node, are "parent" nodes having a plurality of "child" nodes stemming therefrom in the multicast tree. Data 34, such as cells, packets or frames, flow from the root node 20 to the branching point nodes 22, 24, and then to the leaves 26-32. Feedback updates 36 flow from the leaves to the branching point nodes in accordance with a point-to-point flow control technique as is known in the art. While illustrated with a single level of branching point nodes, the actual implementation may have multiple levels of branching point nodes. It should also be understood that the branching point nodes may feed either leaves, as illustrated, or other branching point nodes or combinations thereof depending upon the topology of the multicast tree.

Referring to Figs. 2 and 3, at each branching point node, the forwarding buffer may be modeled as a ring buffer 35 with two pointers: a head pointer 37 and a tail pointer 39. An order retaining buffer system such as the FIFO 16 is also provided and data units are stored in the buffer system upon receipt before entering the ring buffer 35. The tail pointer 39 points to the first received and buffered data unit in the

series of the most recently received data units from upstream that has not yet been forwarded to any branch. The head pointer 37 points to the oldest data unit that needs to be forwarded downstream to any branch. Thus, the data units stored in the ring buffer between the head and tail pointers need to be forwarded to one or more branches. The tail pointer advances by one buffer in a counter-clockwise direction with the forwarding of the most recent data unit in the ring and the arrival of the next most recent data unit that has not yet been forwarded to any branch. The head pointer advances as needed to indicate the current oldest data unit in the ring buffer. The advancement of the head pointer depends on the particular forwarding technique chosen from a range of possible techniques, each of which provides a different service guarantee. Each buffer, i, in the ring has an associated counter called a reference count, r(i), that counts the number of branches to which the data unit in the buffer must be forwarded. If the index (i) increases from head pointer to tail pointer, then one may observe that r(i) is a monotonically increasing function. Each time the tail pointer moves to a new buffer, the corresponding reference count is set to n, the number of downstream branches being serviced by that reference counter. Each time the data unit is forwarded to one of those branches, the reference counter is decremented by one. When the reference counter corresponding to the data unit buffer at

the position of the head pointer reaches 0, the head pointer is advanced in the counter-clockwise direction to the next data unit buffer with a non-zero reference counter. Under no circumstance is the head pointer advanced beyond the tail pointer.

In an alternative embodiment, there can be multiple instances of head and tail pointers with their associated reference counters where each instance can service a different disjoint subset of branches each with a possibly different service guarantee. In this case, it is important to only advance the tail pointer in relation to the head pointers from other subsets to avoid violating the service guarantee associated with those subsets of branches.

In yet another embodiment, there can be a per-branch counter rather than a per-data unit buffer counter. The general principles of the mechanism remain the same.

The root node 20 as well as the branching point nodes 22, 24 execute the multicast forwarding technique. Any one of a plurality of forwarding techniques may be employed at the branching point nodes to support service guarantees for the branches of the multipoint connection. The forwarding techniques function to control the forwarding of multicast data units, and may protect portions of the multicast tree from the effects of poorer performing portions. Forwarding techniques may include a Prevent-Loss (PL) technique, a Prevent-Loss for

Distinguished Subsets (PL(n)) technique, a Prevent-Loss for Variable Subset (p/n) technique, a K-Lag technique, a K-Lead technique and techniques offering other service guarantees.

Prevent-Loss (PL) is a forwarding technique where data unit loss is prevented by adjusting transmission of multicast data units so that each branch receives a copy of each multicast data unit. Prevent-Loss is realized by ensuring that the tail pointer never overtakes the head pointer in all circumstances. If the tail pointer is one data unit buffer position behind the head pointer in the counter-clockwise direction (the ring is full), the data unit at the tail pointer may not be forwarded to any branch. This data unit may be forwarded only when the head pointer advances, i.e., when the reference count of the data unit at the head pointer becomes zero. Depending on the duration of such poor performance, the effect may propagate towards the root of the multicast tree, eventually affecting all leaves.

The Prevent-Loss for Distinguished Subsets technique guarantees delivery of the multicast data unit to a predetermined subsets of branches in the multicast connection. More particularly, transmission to a distinguished subset of branches is made according to the Prevent-Loss technique described above. Hence, there is no data loss in this distinguished subset. A non-distinguished subset of branches consisting of the remaining branches in the connection, is not

guaranteed delivery of the multicast data unit. Hence, the distinguished subsets of branches are insulated from possible poor performance by branches in the non-distinguished subset of branches.

The Prevent-Loss for Distinguished subsets technique is implemented by using a separate set of reference counters for each distinguished subset of branches. The head pointer is advanced only after all members of the distinguished subset have been served. This makes it possible, for example, to provide a lossless service to the members of the distinguished subset of branches. Further, it should be understood that there may be more than one distinguished subset.

In the Prevent-Loss for Variable Subset (p/n) forwarding technique there is no data loss for p out of n branches. This is achieved as a variant to the PL technique. In this technique, the tail pointer is allowed to advance beyond the head pointer, as long as the corresponding reference counter is less than n-p. This ensures that a given data unit copy is delivered to at least p branches. It is possible that the unincluded branch may overflow at some point. However, the performance of the possibly overflowing branch does not affect the performance of branches in the subset of p branches. Hence, the technique provides protection for a subset of branches where the members of the subset are determined contemporaneously with the forwarding calculation.

One embodiment of the Variable Subset technique is illustrated in Fig. 4. In a first step 40 an integer p is entered, where p is a selectable input. At iteration n=0, inquiry is made whether p feedback updates have been received from at least p branches as determined in step 42. Fewer feedback updates may be processed if a timeout occurs in step 44 before p feedback updates are collected. More feedback updates may be processed if a group of feedback updates pushing the total above p arrives contemporaneously. A processed update X(0) is then calculated in step 46 as the median of the gathered feedback updates. A variance v(0) is then calculated in step 50 for use as described below.

A FIFO queue 51 is maintained in the switch for each branch in the multicast connection. The fullness of such FIFO queues is indicative of absence of feedback updates from the associated branches. At an iteration n=m, feedback updates are obtained from p leaves. If at least one leaf fails to deliver a feedback update in the previous iteration (n-1) as determined in step 52, fewer than p leaves are utilized by adjusting the number of leaves required in step 54. Given the following definitions:

q(i) = fullness of forwarding FIFO queue i branch;

q(j,m) = fullness of forwarding FIFO queue j branch at iteration m;

$x(m)$ = median of $q(i,m)$ at iteration m;

$v(m)$ = variance of $q(i,m)$ at iteration m;

$x(m,m-1)$ = median of $q(i,m)$ over iterations m and m-1; and

$v(m,m-1)$ = variance of $q(i,m)$ over iterations m and m-1, then

an outlier (j) at time (n=m) is removed from the pool and is not to be waited for at n=m+1 if the following is true:

$[q(j,m) - x(m)]^2 > v(m)$ or

$[q(j,m)-x(m,m-1)^2 > v(m,m-1)$, where

$v(m)=\sum_i[((q(i)-x(m))/p]^2$,

$v(m-1,m-2) = \sum_i \sum_j [((q(i)q(j) - x(m-1, m-2))/p]^2$,

$x(m)$ = median of p updates, and

$x(m-1, m-2)$ = median of p updates at both time instances.

However, if a feedback update is received from the previously silent node while updates are being gathered, then that feedback update is included in the calculation. Hence, $v(m)$ is calculated in step 50, and any branches which have not provided a feedback update in the previous iteration should not be considered in the next iteration are detected in step 52.

If all leaves were unresponsive in the previous "y" iterations, where "y" is a small number like 1, 2 or 3, then the processed update is computed with fewer than p updates. If an update is received from a specified leaf, that update is

added to the pool for processing. Further, if the outbound FIFO for a specified leaf or branch is stale (i.e., if that FIFO corresponds to a branch that is not being considered), then four steps may be executed: (A) drop the p earliest data units in the FIFO; or (B) forward the p earliest data units in the FIFO; or (C) drop the p latest data units in the FIFO; or (D) forward the p latest data units and drop the remaining data units in the FIFO. In cases C and D, the FIFO for the specified leaf is considered to be RESTARTED.

Referring to Figs. 2 and 5, in the K-Lag service, the slowest branches are guaranteed to not lag the fastest by more than K data units. The K-Lag service is realized by ensuring that the head pointer is advanced together with the tail pointer such that it is no more than a distance of K from the tail pointer. If the tail pointer is K data unit buffer positions ahead of the head pointer in the counter-clockwise direction and if the data unit at the tail pointer has been forwarded to any branch, then the tail pointer is advanced along with the head pointer one data unit buffer position counter-clockwise. In this circumstance the data unit in the buffer position of the original head pointer is no longer guaranteed to be forwarded. It is possible that the data unit at the previous head pointer positions may yet be forwarded before being overwritten by that at the tail pointer. In an alternative embodiment, such a data unit can be deleted and not

forwarded any more. It should also be noted that K must be at least 1. Different values of K give rise to different levels of service.

In a variation of the K-Lag technique, K-EPD, each branch is allowed to lag by up to K data units, as set by an upper memory bound associated with the leaf. If the tail pointer is K data unit buffer positions ahead of the head pointer in the counter-clockwise direction and the head pointer is pointing to a data unit in the middle of the frame, then the head pointer is advanced up to the end of the frame data unit or one position before the tail pointer, to prevent forwarding of all data units associated with the respective frame and thus avoid the waste of network bandwidth and resources.

In the K-Lead technique, the fastest branches cannot be ahead of any other branches by more that K data units. The K-Lead technique is realized by ensuring that the tail pointer is not advanced more that a distance of K from the head pointer. If the tail pointer is K data unit buffer positions ahead of the head pointer in the counter-clockwise direction, the data unit at the tail pointer may not be forwarded to any branch.

Having described the preferred embodiments of the invention, it will now become apparent to one of skill in the art that other embodiments incorporating the presently disclosed method and apparatus may be used. Accordingly, the invention should not be viewed as limited to the disclosed

embodiments, but rather should be viewed as limited only by the spirit and scope of the appended claims.

CLAIMS

1. A method for forwarding a multicast data unit from a branching node to at least one of a plurality of coupled downstream nodes, comprising the steps of:

determining, at said branching point node, the eligibility of respective downstream nodes to receive said multicast data unit based upon a specified multicast data unit forwarding technique selected from a plurality of multicast data unit forwarding techniques; and

forwarding said data unit to selected ones of said coupled downstream nodes based upon said specified data unit forwarding technique.

2. The method of claim 1 wherein said eligibility determining step includes analyzing feedback updates received from respective downstream nodes.

3. The method of claim 1 including the further step of employing a data unit forwarding ring buffer to execute said specified data unit forwarding technique.

4. The method of claim 1 wherein the processing step includes the further step of employing a data unit forwarding technique in which no downstream node is allowed to lead ahead of another downstream node by more than K data units.

5. The method of claim 1 wherein the processing step includes the further step of employing a data unit forwarding technique in which loss of data is prevented between the branching node and each downstream node in the multicast connection.

6. The method of claim 1 wherein the processing step includes the further step of employing a data unit forwarding technique in which loss of data units is prevented between the branching point node and a distinguished subset of downstream nodes in the multicast connection.

7. The method of claim 6 including the further step of designating a non-distinguished subset of downstream nodes in the multicast connection which are not entirely protected from data loss.

8. The method of claim 1 wherein the processing step includes the further step of employing a technique in which loss of data is prevented between the branching node and a variable subset of (p) downstream nodes in the multicast connection having most recently provided a feedback update to the branching node.

9. The method of claim 8 including the further step of including at least the (p) most recent feedback updates to determine the downstream nodes in the distinguished subset, where (p) is an input.

10. The method of claim 1 wherein the processing step includes the further step of employing a data unit forwarding technique in which no downstream node is allowed to lag behind another downstream node by more than K data units.

11. The method of claim 10 wherein the data units are grouped into frames, and including the further step of discarding any remaining portion of a frame if a downstream node lags behind another downstream node by K data units while the frame is being transmitted.

12. A multicast apparatus for transmitting multicast data units in a multicast connection from a branching node to a plurality of downstream nodes, comprising:

a memory in each downstream node for receiving multicast data units;

a circuit for providing information indicating fullness of the downstream node memory, the circuit providing such fullness information to the branching node in accordance with a point-to-point flow control technique; and

a circuit in each branching node for processing the fullness information in accordance with a data unit forwarding technique selectable from a plurality of data unit forwarding techniques.

13. The apparatus of claim 12 further including a circuit in each downstream node for providing feedback updates indicating memory fullness to the branching node.

14. The apparatus of claim 12 wherein the branching point node includes a data unit forwarding ring buffer having a head pointer and a tail pointer.

15. The apparatus of claim 14 wherein the branching point node includes a buffer system for storing incoming data units ring buffer prior to forwarding of such incoming data units to the ring buffer.

16. The apparatus of claim 12 wherein the selected data unit forwarding technique prevents any downstream node from leading ahead of any other downstream node by more than K data units.

17. The apparatus of claim 12 wherein the selected data unit forwarding technique prevents loss of data units between the branching node and each downstream node connected thereto.

18. The apparatus of claim 12 wherein the selected data unit forwarding technique prevents loss of data units between the branching point node and a distinguished subset of downstream nodes connected thereto.

19. The apparatus of claim 18 wherein the selected data unit forwarding technique does not prevent loss for a non-distinguished subset of downstream nodes connected thereto.

20. The apparatus of claim 12 wherein the selected data unit forwarding technique prevents data unit loss for a variable subset (p) of downstream nodes connected thereto.

21. The apparatus of claim 20 wherein the members of the variable subset (p) are the (p) downstream nodes most recently having provided memory fullness information to the branching node, where (p) is an input.

22. The apparatus of claim 12 wherein the selected data unit forwarding technique prevents any downstream node from lagging behind any other downstream node by more than K data units.

23. The apparatus of claim 22 wherein the data units are grouped into frames, and wherein any portion of a frame remaining to be transmitted is discarded when a downstream node

lags behind another downstream node by K data units while the frame is being transmitted.

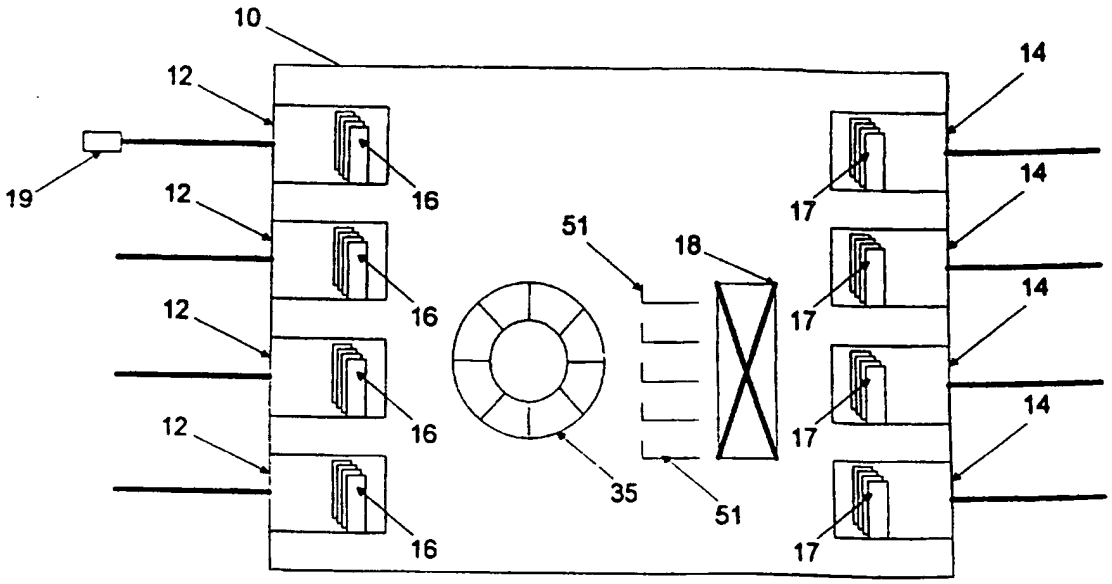


Fig. 1

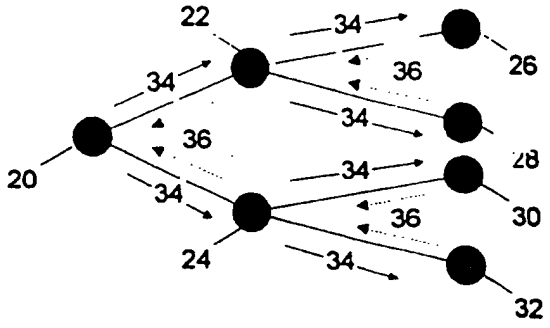


Fig. 2

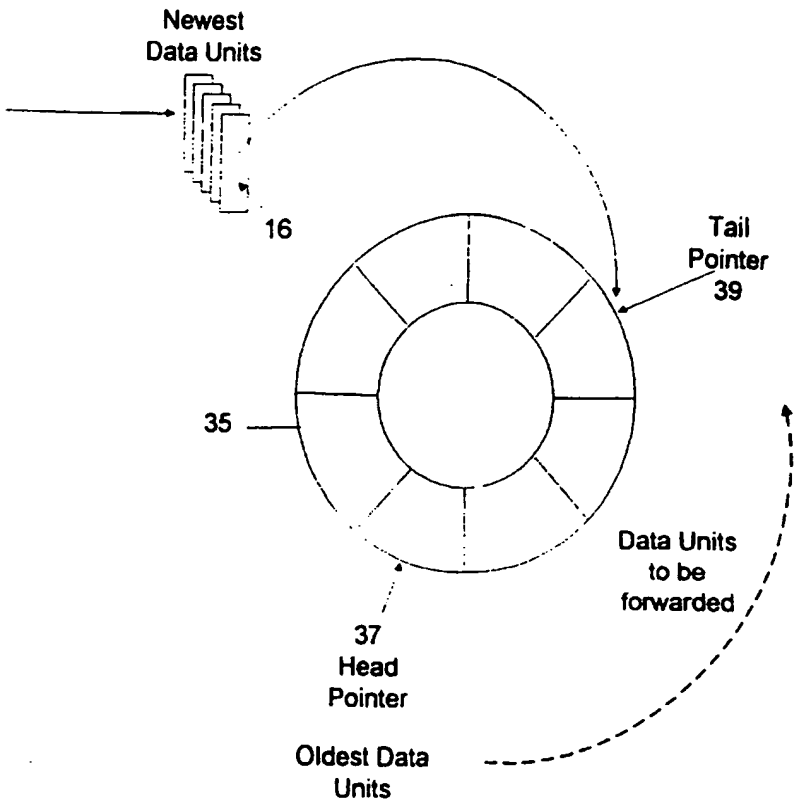

Newest
Data Units
16
Tail
Pointer
39
35
Data Units
to be
forwarded
37
Head
Pointer
Oldest Data
Units

Fig. 3

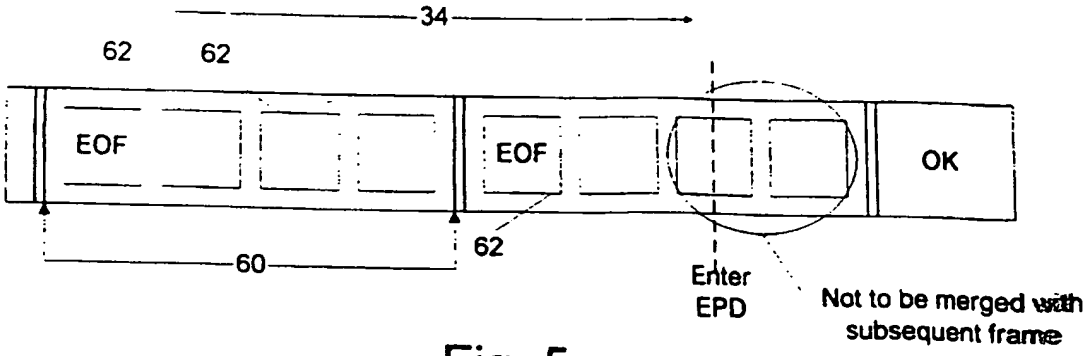

34
62
62
EOF
EOF
OK
62
60
Enter
EPD
Not to be merged with
subsequent frame

Fig. 5

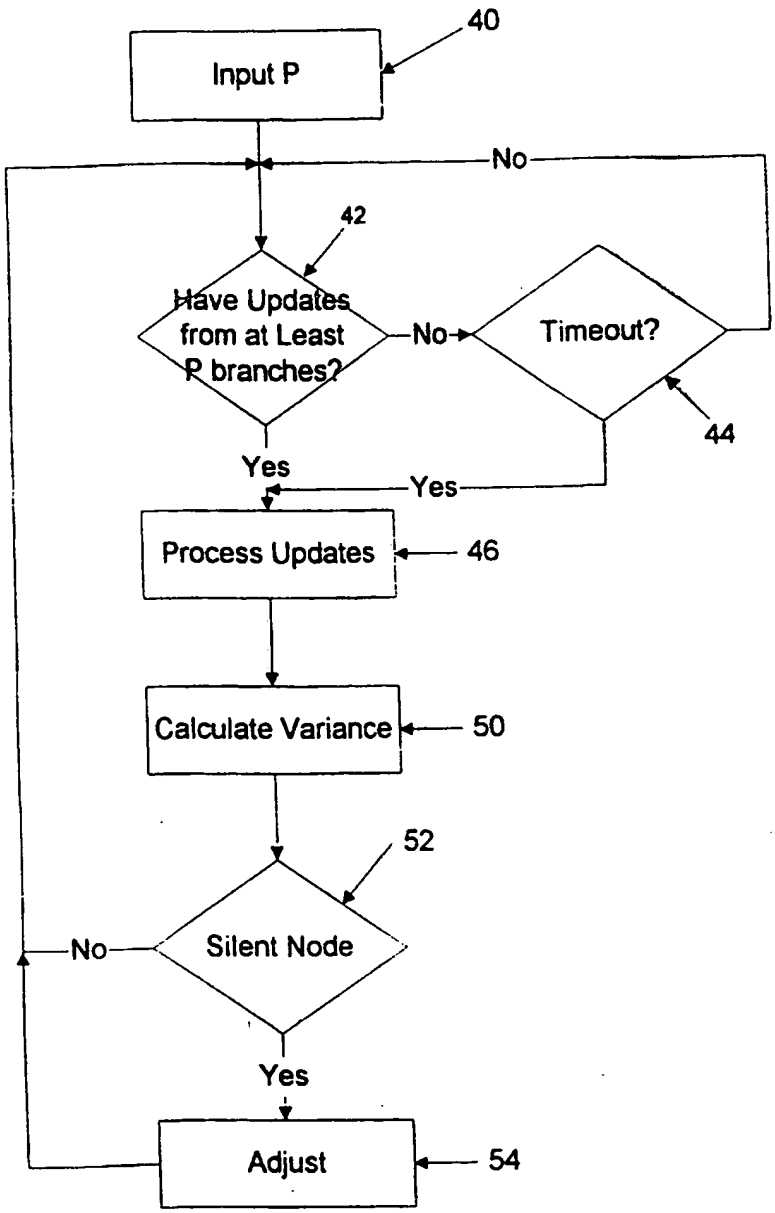

Input P
40
No
42
Have Updates from at Least P branches?
No
Timeout?
44
Yes
Yes
Process Updates
46
Calculate Variance
50
52
Silent Node
No
Yes
Adjust
54

Fig. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/00488

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :H04L 12/18, 12/54
US CL :370/235, 390, 412, 413, 429, 432
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 370/235, 390, 412, 413, 429, 432

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS
search terms: multicast, tree, branch

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,570,348 A (HOLDEN) 29 October 1996, column 7, line 60 to column 9, line 7.	1-23
A,P	US 5,517,494 A (GREEN) 14 May 1996, column 2, lines 18-37.	1-23

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be part of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
02 MARCH 1997	26 MAR 1997
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer MELVIN MARCELO Telephone No. (703) 305-4700

Form PCT/ISA/210 (second sheet)(July 1992)*

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.